

REMARKS

Applicants have filed the present continuation in furtherance of the prosecution of subject matter not previously allowed in the parent application. In this response, Applicants have amended the specification, cancelled claims 7-29 and 39-42, amended claims 1-6 and 30-38, and added new claims 42-45. Therefore, claims 1-6, 30-38, and 42-45 are currently pending. To reduce the potential for duplicative rejections by the Examiner, Applicants have chosen to respond to the Examiner's previous rejections herein. Accordingly, Applicants respectfully request entry of this Preliminary Amendment and allowance of the present claims in view of the remarks set forth below.

Amendment to the Specification

Applicants have amended the specification to reference U.S. Patent Application Serial No. 09/303,799, which is the parent to the present application.

Amendments to Claims

In U.S. Patent Application Serial No. 09/303,799, Applicants amended claims 1-6 and 30-38 to clarify the subject matter being claimed and to comply with M.P.E.P. § 608.01(v). Specifically, claims 1-6 and 30-38 have been amended to place them in a form similar to the claims as rejected.

Rejections Under 35 U.S.C. § 103

In the Official Action mailed on September 11, 2002, the Examiner rejected claims 1-6, 30-34, 35-38, and 43-45 under 35 U.S.C. § 103(a) as being unpatentable over Borgendale et al. (U.S. Pat. No. 5,774,720) in view of "Java 2 Platform, Standard Edition, v1.2.2 API Specification; Class window." Specifically, in the rejection, the Examiner stated:

Claims 1-6, 30-34, 35-38 and 43-45 are rejected under 35 U.S.C. 103(a) as being unpatentable over Borgendale et al, U.S. Patent No. 5,774,720 in view of “Java 2 Platform, Standard Edition, v1.2.2 API Specification; Class window” (“Java 2 Platform”), Art of record.

As per claim 1, Borgendale et al disclose that **creating a first window in a native application, sending a command for a second window to the first window, passing the command from the first window to the second window through a native interface** (abstract, “...The graphics management system includes a common graphical user interface (GUI) which receives graphics related application programming interface (API) calls from the application programs, and transforms the various types of graphics API calls native to a particular GUI, into a generic format compatible with a personality neutral graphics engine. The personality neutral calls are then passed from the CGUI to the personality neutral graphics engine which services the calls, and controls the drawing of lines, circles and other drawing tasks for each of the windows presented on the display...” and Col. 5 lines 55 to 64; Note that first window is interpreted as windowing API function calls and the second window is interpreted as the windows controlled by the personality neutral graphics engines, and the native interface is the CGU).

Borgendale et al doesn't explicitly disclose the command is a JAVA command and the second window is a **Java window**.

However, “Java 2 Platform” discloses Java commands and a Java window (p. 1).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the teaching of “Java 2 Platform” into the system of Borgendale et al to have the command to be a JAVA command and the second window to be a Java window because it provides an efficient method to adapt the teaching of Borgendale into various systems with various types of commands and various types of window, including Java command and Java windows to meet the various needs of the systems. Doing so facilitating the passing of various types of commands native to a particular system to a Java window.

In the Response to Office Action submitted on December 16, 2002, Applicants first argued that the Borgendale reference and the Java 2 Platform reference did not disclose “a

JAVA window” and “passing the command from the first window to the JAVA window through a native interface.” Applicants secondly argued that the references lacked the motivation or suggestion for the combination, and, as such, failed to support an obviousness rejection. In the Final Official Action mailed on February 11, 2003, the Examiner repeated the rejection, and in the Response to Arguments, the Examiner stated:

- I) Borgendale discloses translating or converting commands, but the present invention disclose passing untranslated commands between the first and Java window.

Examiner’s response:

- I) In response to applicant’s argument that the references fail to show certain features of applicant’s invention, it is noted that the features upon which applicant relies (i.e., untranslated commands) are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. *See In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

- II) Nothing in “Java 2 platform” discloses sending a Java command to a Java window from a first window through a native interface.

Examiner’s response:

- II) “Java 2 platform” is cited only to disclose Java window and Java command are well known. Borgendale discloses sending command to a window from a first window through a native interface. The previous office action disclose why it would have been obvious to combine Borgendale and Java 2 platform. Applicant failed to point out the error in the motivation. Therefore, the rejection is maintained.

Applicants again respectfully traverse this rejection. The burden of establishing a *prima facie* case of obviousness falls on the Examiner. *Ex parte Wolters and Kuypers*, 214 U.S.P.Q. 735 (PTO Bd. App. 1979). Obviousness cannot be established by combining the teachings of the prior art to produce the claimed invention absent some teaching or suggestion supporting the combination. *ACS Hospital Systems, Inc. v. Montefiore Hospital*, 732 F.2d 1572, 1577, 221 U.S.P.Q. 929, 933 (Fed. Cir. 1984). Accordingly, to establish a

prima facie case, the Examiner must not only show that the combination includes *all* of the claimed elements, but also a convincing line of reason as to why one of ordinary skill in the art would have found the claimed invention to have been obvious in light of the teachings of the references. *Ex parte Clapp*, 227 U.S.P.Q. 972 (B.P.A.I. 1985). When prior art references require a selected combination to render obvious a subsequent invention, there must be some reason for the combination other than the hindsight gained from the invention itself, i.e., something in the prior art as a whole must suggest the desirability, and thus the obviousness, of making the combination. *Uniroyal Inc. v. Rudkin-Wiley Corp.*, 837 F.2d 1044, 5 U.S.P.Q.2d 1434 (Fed. Cir. 1988).

In the present application, independent claims 1, 30, and 35 each recite various unique features of the present application. For instance, claim 1 recites “sending a command for a JAVA window to the first window,” and “passing the command from the first window to the JAVA window through a native interface.” Similarly, claim 30 recites “means for sending a command for the JAVA window to the first window” and “means for passing the command from the first window to the JAVA window through a native interface.” Claim 35 recites “code to send a command for a JAVA window to the first window” and “code to pass the command from the first window to the JAVA window through a native interface connection.”

Conversely, the Borgendale reference describes a system for allowing application programs that are written for different graphical interfaces to be executed on a single desktop. See Borgendale et al., col. 3, lines 12-18. The reference discloses that different graphical user interfaces (GUIs) present compatibility issues because each GUI has its own application programming interface and functions that are limited to an operating system. See Borgendale et al., col. 2, lines 17-30. The advantage of the Borgendale et al. system is that a single graphical user interface and operating system are used to execute application programs that

are written for different types of GUIs. *See Borgendale et al.*, col. 4, lines 6-21. To provide a single graphical interface, Application programs 82 issue API calls to a common graphical unit interface (CGUI) 86. *See Borgendale et al.*, col. 5, lines 39-49. Through the use of a personality neutral graphics manager 90, the CGUI 86 translates the different API calls into a format compatible with the graphic engine 100, which draws the lines, circles, and points in the display 42. *See Borgendale et al.*, col. 5, lines 49-51; col. 6, lines 7-14. Also, with a personality neutral event/window manager 88, the CGUI 86 translates the API calls to create, destroy, and process the windowing related API calls. *See Borgendale et al.*, col. 5, lines 55-64. Thus, the Borgendale reference teaches using the personality neutral event/window manager 88 and the personality neutral graphics manager 90 to manage the windows.

In the rejection of the claims, the Examiner discussed independent claim 1 and applied those arguments to the other independent claims 30 and 35. The Examiner asserted that the Borgendale reference taught all of the recited features, except the command being a JAVA command and the second window being a JAVA window. For this recited feature, the Examiner relied on the Java 2 Platform reference, which is a JAVA API specification, to teach this subject matter. In the rejection, the Examiner specifically asserted that windowing API function calls are equivalent to the “first window,” the windows controlled by the personality neutral graphic engines are the “second window,” and the CGUI is the “native interface.”

Despite these assertions, the Examiner’s rejection of independent claims 1, 30, and 35 fails for at least two reasons. First, the Borgendale and Java 2 Platform references fail to include all of the elements recited within the claims. For instance, the references fail to disclose or teach “sending a command for a JAVA window to the first window” and “passing

the command from the first window to the JAVA window through a native interface,” as recited in claims 1 and 30. Also, the references fail to disclose or teach “code to send a command for a JAVA window to the first window” and “code to pass the command from the first window to the JAVA window through a native interface connection,” as recited in claim 35. Secondly, the Borgendale reference and the Java 2 Platform reference do not provide or support the combination suggested by the Examiner. Hence, to the extent that the Examiner is relying upon the combination of these cited references, the rejection cannot stand.

With regard to the first reason, in applying the Borgendale reference, the Examiner appears to have misinterpreted the reference and misapplied it to the presently recited claims. As noted above, the Examiner asserted that windowing API function calls are equivalent to the “first window.” However, the API function calls are merely commands that are issued from Application programs 82 to the personality neutral event/window manager 88 and to the personality neutral graphic manager 88, which are parts of the CGUI 86. These windowing API function calls are translated by the personality neutral event/window manager 88 to create, destroy, and process the windowing related API calls, while the graphic API function calls are translated by the personality neutral graphic manager 90 to draw lines and circles. *See Borgendale et al., Fig. 3; col. 5, lines 55-64.* As such, the windowing API function calls are not windows, but commands that comply with a specific format.

In view of the Examiner’s misinterpretation and the actual teachings in the Borgnedale reference, the Borgendale reference clearly fails to disclose or teach “sending a command for a JAVA window to the first window,” as recited in claims 1 and 30, and “code to send a command for a JAVA window to the first window,” as recited in claim 35. As such, for the rejection to stand, the Java 2 Platform reference must provide the missing

subject matter. However, the Java 2 Platform reference does not cure the deficiencies of the Borgendale reference. The Java 2 Platform reference is merely a specification that relates to Java 2 platform APIs. The Java 2 Platform reference merely discloses JAVA commands and other information relating to the APIs. As such, the Java 2 Platform reference clearly does not disclose or teach “sending a command for a JAVA window to the first window,” as recited in claims 1 and 30, or “code to send a command for a JAVA window to the first window,” as recited in claim 35.

Furthermore, the Borgnedale reference fails to disclose or teach “passing the command from the first window to the JAVA window through a native interface,” as recited in claims 1 and 30, and “code to pass the command from the first window to the JAVA window through a native interface connection,” as recited in claim 35. As discussed above, the Borgendale reference describes translating API calls being sent to the CGUI 86 from Application programs 82. *See* Borgendale et al., Fig. 3; col. 5, lines 39-64. The CGUI 86 uses this information to present the windows on the display 45. *See* Borgendale et al., col. 5, line 55 to col. 6, line 15. However, nothing in the reference discloses or teaches that the windows, such as windows 50 and 52, communicate with each other through the CGUI 86. *See* Borgendale et al., Fig. 2. As such, for the rejection to stand, the Java 2 Platform reference must provide the missing subject matter.

Again, the Java 2 Platform reference does not cure the deficiencies of the Borgendale reference. The Java 2 Platform reference is simply a specification that is used by the Examiner to provide a JAVA window. Thus, the Java 2 Platform reference clearly does not disclose or teach “passing the command from the first window to the JAVA window through

a native interface,” as recited in claims 1 and 30, or “code to pass the command from the first window to the JAVA window through a native interface connection,” as recited in claim 35.

With regard to the second reason, the Examiner has failed to point to a convincing suggestion or teaching that would motivate one skilled in the art to modify the Borgendale reference or the Java 2 Platform reference. Indeed, the Examiner has failed to meet the required burden of providing *evidence* of a motivation for the combination of Borgendale et al. and Java 2 Platform reference. Instead, the Examiner stated that:

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the teaching of “Java 2 Platform” into the system of Borgendale et al to have the command to be a JAVA command and the second window to be a Java window because it provides an efficient method to adapt the teaching of Borgendale into various systems with various types of commands and various types of window, including Java command and Java windows to meet the various needs of the systems. Doing so facilitating the passing of various types of commands native to a particular system to a Java window.

This statement is nothing more than an unsupported assertion about the teachings of Borgendale and Java 2 Platform, not a convincing line of reasoning supported by actual evidence as to *why* one of skill in the art would combine the references.

The Federal Circuit recently overturned the Board, which had upheld an examiner’s rejection in a similar situation. In the case of *In re Lee*, 61 U.S.P.Q.2d 1430 (Fed. Cir. 2002), the Examiner rejected the applicant’s claims under 35 U.S.C. § 103 without providing evidence of motivation to combine the references. The Board subsequently affirmed the examiner’s rejection. In overturning the Board’s decision, the Federal Circuit stated that:

When patentability turns on the question of obviousness, the search for and analysis of the prior art includes evidence relevant to the finding of whether there is a teaching, motivation, or suggestion to select and combine the

references relied on as evidence of obviousness. *See, e.g., McGinley v. Franklin Sports, Inc.*, 262 F.3d 1339, 1351-52, 60 U.S.P.Q.2d 1001, 1008 (Fed. Cir. 2001) (“the central question is whether there is reason to combine [the] references,” a question of fact drawing on the Graham factors).

‘The factual inquiry whether to combine references must be thorough and searching.’ *Id.* It must be based on *objective evidence of record*. This precedent has been reinforced in myriad decisions, and cannot be dispensed with. [citations omitted]. *In re Lee*, 61 U.S.P.Q.2d at 1433 (emphasis added).

Accordingly, the Examiner’s unsupported assertion does not meet the evidentiary standard required for combining references under Section 103. Indeed, the Examiner has, at best, impermissibly relied on hindsight by using the teachings of Applicants to find the suggestion to combine the alleged teachings of Borgendale et al. and Java 2 Platform reference.

As noted above, the Borgendale reference discloses that different graphical user interfaces (GUIs) present compatibility issues because each GUI is specific to the application that it interacts with. *See* Borgendale et al., col. 2, lines 17-21. The application programming interface communicate with a GUI by calls implemented by the GUI. *See* Borgendale et al., col. 2, lines 22-27. As such, the Application is limited to interacting with only that GUI. *See* Borgendale et al., col. 2, lines 27-30. The alleged advantage of the Borgendale et al. system is that it uses a single graphical user interface and operating system to execute application programs that are written for different types of graphical user interfaces. *See* Borgendale et al., col. 4, lines 6-21. In fact, the reference is devoid of any reference or teachings relating to JAVA. In contrast, the Java 2 Platform reference is merely a specification that discloses JAVA commands and other information relating to the APIs. Accordingly, each of the references clearly fails to disclose or suggest any desirability for the proposed combination of the references. In addition, the Examiner has not even pointed to any suggestion in the

references or in any other art of record for the proposed combination. Thus, the Examiner's proposed combination of the Borgendale et al. and Java 2 Platform specification is not supported.

In view of the remarks set forth above, Applicants respectfully submit that subject matter disclosed in the Borgendale and Java 2 Platform references do not render the claimed subject matter obvious. Therefore, Applicants respectfully request allowance of the pending independent claims 1, 30 and 35, and the respective claims depending therefrom.

Conclusion

In view of the remarks and amendments set forth above, Applicants respectfully request allowance of all pending claims 1-6, 30-38 and 43-45. If the Examiner believes that a telephonic interview will help speed this application toward issuance, the Examiner is invited to contact the undersigned at the telephone number below.

Respectfully submitted,

Date: September 16, 2003


Michael G. Fletcher
Reg. No. 32,777
(281) 970-4545

CORRESPONDENCE ADDRESS:

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
Legal Department, M/S 35
P.O. Box 272400
Fort Collins, Colorado 8-527-2400